ANIME GENERATION USING GENERATIVE ADVERSARIAL NETWORKS (GANs)

BY

UTSAV PATHAK (19030141CSE060)

Semester: V

Department of Computer Science and Engineering

Alliance College of Engineering and Design

ALLIANCE UNIVERSITY

Chikkahagade cross, Chandapura- Anekal Main Road Bangalore-562106

Batch: 2019-23

INDEX

ABSTRACT

With advancements in artificial intelligence and machine learning techniques, we are now at a stage and advancing more into it, making it reach a point when we are trying to solve more complex problems, it has conventionally happened over time that the collected data is not as diverse as it should be and with time it becomes difficult to find image data. This has created a problem, a problems that enables models not to be trained well enough for existing scenarios.

Referring to Style GANs paper, we have tried to implement the generation of Anime using existing dataset on Kaggle and use mapping network techniques on input vectors in order to create intermediatory latent space. We therefore are able to obtain the data and correlations with different directional outlooks and vectors and hence are able to have an overall view of how better the output is in comparison to the existing data and how accurate our generations are. Existing solutions have been able to create an accuracy of about 86% on their models, while setting input vectors and increasing their concentration to upper edges of the face and retraining the model sufficiently enough, we have obtained a cut-up accuracy of 91% on our model. We believe that TensorFlow has helped us in receive better results on vectoring and thereby increasing both the quality and quantity of our output.

The discriminator takes an image as input, and tries to classify it as "real" or "generated". In this sense, it's like any other neural network. We'll use a convolutional neural networks (CNN) which outputs a single number output for every image. We'll use stride of 2 to progressively reduce the size of the output feature map.

Being an extension to GANs, Style GANs provide a wider and more interactive approach to the data being trained and the way the generator model traditionally works. In Style GAN, we change the generator model in such a way that we use intermediate latent space in each point in the generators to control the output of the model and the variation in sources at each point of the network formed.

Traditionally, we do not have much control over the generation and intermediatory details of generation points in the GANs, but using style GAN for this particular purpose, gives a completely new direction to the unsupervised learning methodology in the flow. This readily increases the availability of configurable vectors to the algorithm and therefore increases the flexibility to gain more “knowledge”, this further helps to form a clear cut instruction set and gain more while training.

All these practices makes the overall algorithm power enough to generate better faces with very less noise and more power.

Introduction

Generative Model is an unsupervised machine learning task and therefore, it becomes very crucial to train the model in such a way that we have information and enough amount of vectors to create new data and generate new faces in this particular case.

We'll be using Deep Convolutional Generative Adversarial Networks (DC-GANs) for our project. Though we'll be using it to generate the faces of new anime characters, DC-GANs can also be used to create modern fashion styles, general content creation, and sometimes for data augmentation as well.

GANs typically employ two duelling neural networks to train a computer to learn the nature of a dataset well enough to generate convincing fakes. One of these Neural Networks generates fakes (the generator), and the other tries to classify which images are fake (the discriminator). These networks improve over time by competing against each other. Perhaps imagine the generator as a robber and the discriminator as a police officer. The more the robber steals, the better he gets at stealing things. But at the same time, the police officer also gets better at catching the thief. Well, in an ideal world, anyway. The losses in these neural networks are primarily a function of how the other network performs: Discriminator network loss is a function of generator network quality: Loss is high for the discriminator if it gets fooled by the generator's fake images. Generator network loss is a function of discriminator network quality: Loss is high if the generator is not able to fool the discriminator. In the training phase, we train our discriminator and generator networks sequentially, intending to improve performance for both. The end goal is to end up with weights that help the generator to create realistic-looking images. In the end, we'll use the generator neural network to generate high-quality fake images from random noise.

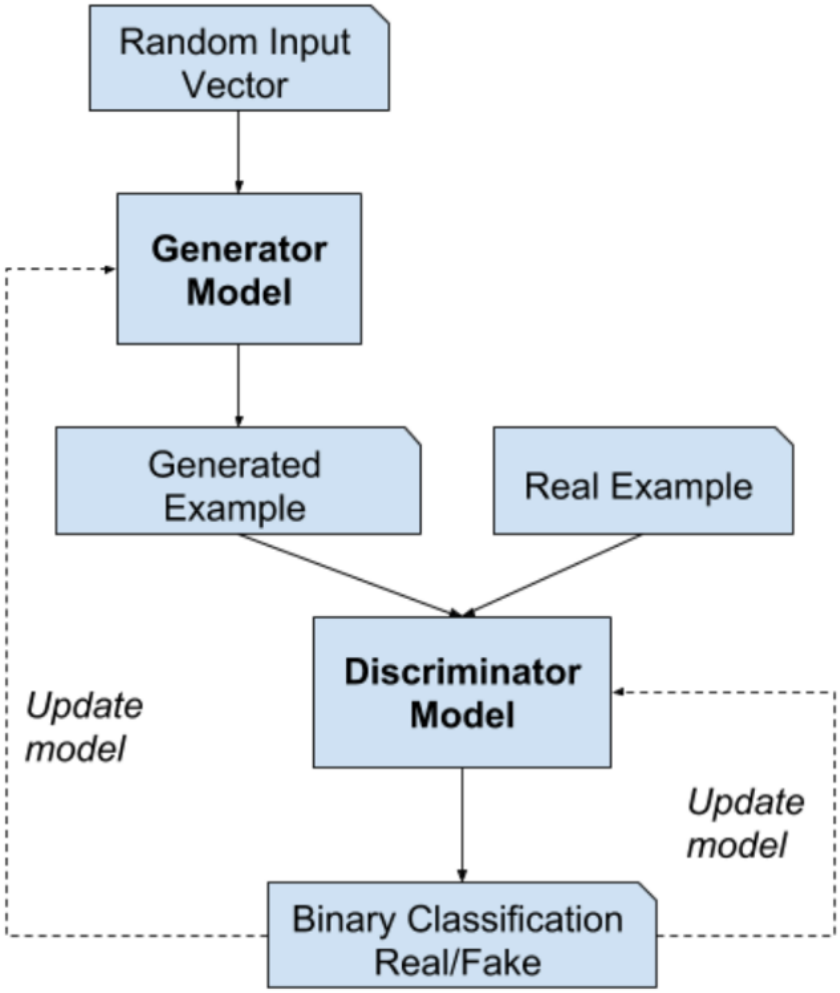
The drawbacks of traditional GANs are covered by StyleGAN and hence we use it for better accuracy and reduced noise in the generated data. In generative adversarial networks, two networks train and compete against each other, resulting in mutual improvisation. The generator misleads the discriminator by creating compelling fake inputs and tries to fool the discriminator into thinking of these as real inputs . The discriminator tells if an input is real or fake.

The reason for combining both networks is that there is no feedback on the generator's outputs. The ONLY guide is if the discriminator accepts the generator's output. The dataset for anime faces can be generated by curling through various manga websites and downloading images, cropping faces out of them, and resizing them to a standard size.

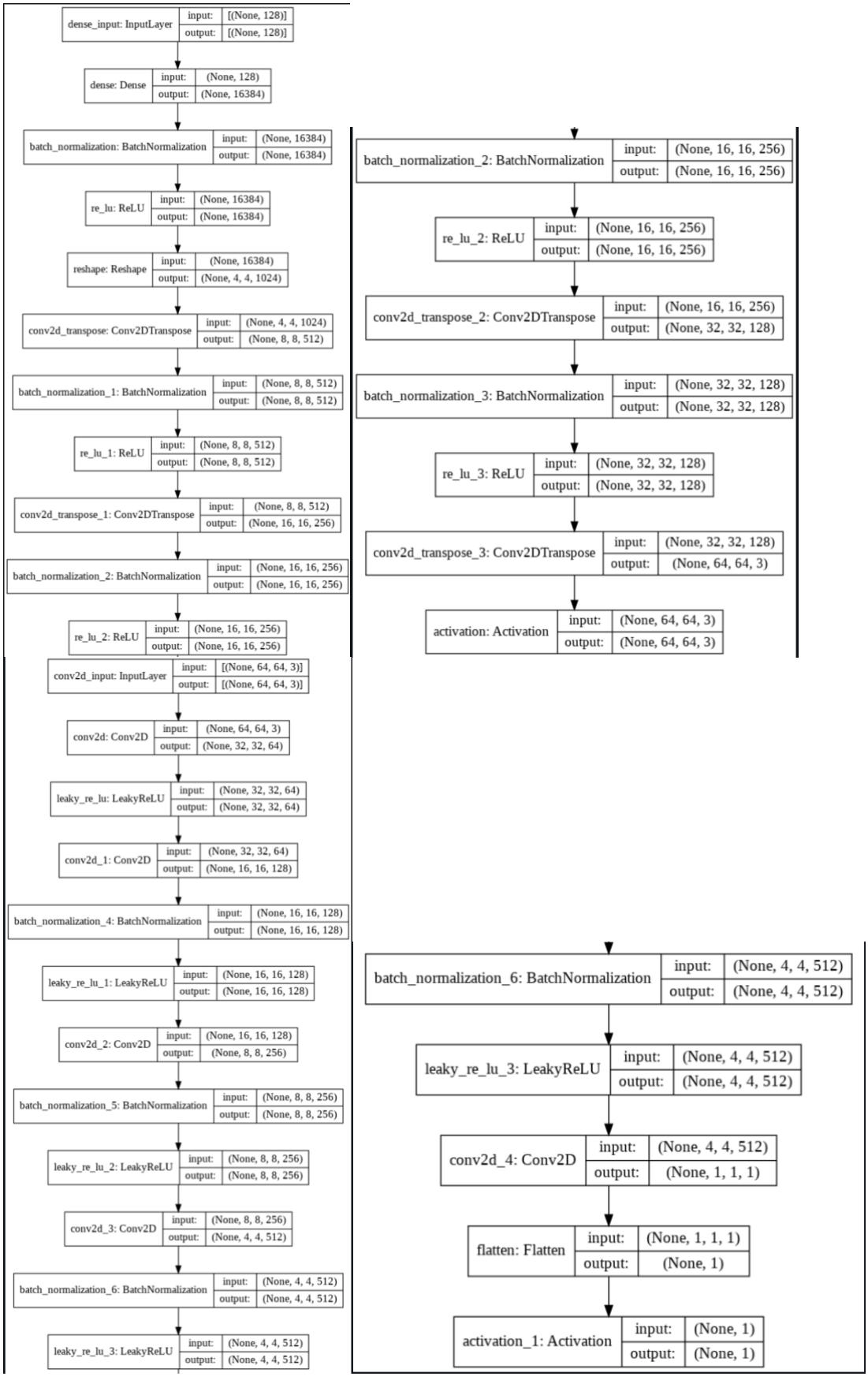
The dataset for training used is available at:
https://www.kaggle.com/aadilmalik94/animecharacterfaces.

The generator consists of convolution transpose layers followed by batch normalization and a leaky ReLU activation function for up sampling. We'll use a strides parameter in the convolution layer. This is done to avoid unstable training. Leaky ReLU functions are one attempt to fix the problem of dying in the activation function. Instead of the function being zero when $x < 0$, a leaky ReLU will instead have a small negative slope.

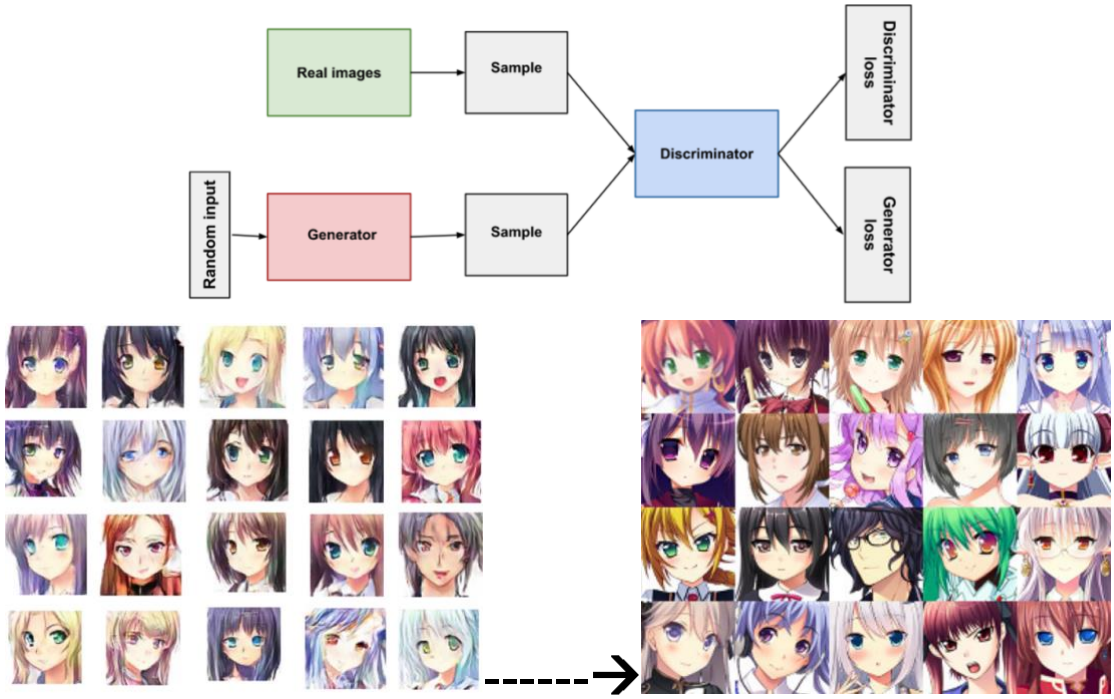
Design Flow


Random Input Vector
Generator Model
Generated Example
Real Example
Discriminator Model
Update model
Update model
Binary Classification Real/Fake

Model Information



Results



As seen in the above two pictures of input and output, the StyleGAN, does beautification on the data and processes the vectors to obtain more dense and sharp images of anime characters.

Conclusion

Conclude the project with better possibilities. The project received an overall accuracy of 91% on comparison with existing performed on Manga Dataset, of 86%. This was obtained by avoiding overtraining and limiting the vectors to only a certain field and particular area in the generative model.

Future Scope

Better works with near ideal accuracy could be done to create human faces and thereby be helpful in training algorithms that demand natural faces, but we need to maintain privacy and therefore we would avoid collecting real data of humans.
This would also help in bridging the gap we have in data, like of underrepresented communities, and hence the models could be more intelligent and hence also reduce discrimination, especially in the fashion industry.

Bibliography

1. GANs issue of THE BATCH from deeplearning.ai

2. Generative Adversarial Nets Ian J. Goodfellow∗ , Jean Pouget-Abadie† , Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair‡ , Aaron Courville, Yoshua Bengio§ Departement d'informatique et de recherche op´ erationnelle´ Universite de Montr´ eal´ Montreal, QC H3C 3J7

3. Improved Techniques for Training GANs Tim Salimans [tim@openai.com] Ian Goodfellow [ian@openai.com] Wojciech Zaremba [woj@openai.com] Vicki Cheung [vicki@openai.com] Alec Radford [alec@openai.com] Xi Chen [peter@openai.com]

Complete Project With Code can be accessed at: https://github.com/pathakutsav/AnimeGenerationUsingStyleGANs